

Le module Turtle

1. PRESENTATION

Le module Turtle (tortue) est un standard sur Python. C'est une adaptation de la tortue Logo (1980, robot possédant un stylo qui était capable de dessiner sur une feuille en papier) qui permettait l'apprentissage de façon ludique de l'algorithmie.

Ce module permet de dessiner des formes géométriques en lui donnant des commandes simples comme avancer (`forward`), tourner à gauche (`left`), etc...

L'objectif ici est de ne pas connaître toutes les instructions qu'offre ce module, elles sont nombreuses, pour cela vous pouvez vous référer au site :

<https://docs.python.org/3/library/turtle.html>, mais d'en connaître quelques-unes afin de réaliser des figures intéressantes d'un point de vue algorithmique.

C'est un module qu'il faut importer, donc pour pouvoir l'utiliser, la première ligne à mettre dans le programme est la suivante :

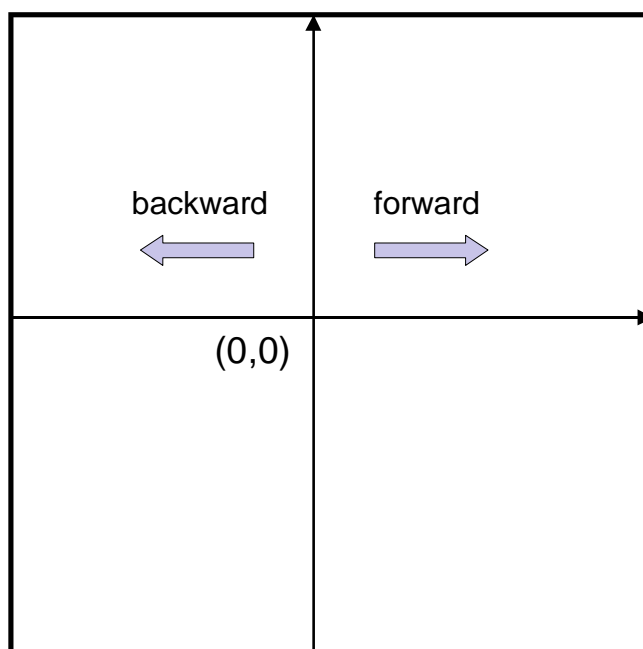
```
from turtle import *
```

Ou

```
import turtle as tu
```

Dans ce dernier cas pour appeler une fonction de la bibliothèque Turtle, il faudra la préfixer `tu` ("tu" ici, car on peut changer de nom).

L'écran est défini ainsi :



Le point de coordonnées (0,0) est au centre de l'écran. En mode standard la direction va vers la droite, le sens de rotation est trigonométrique et les angles sont exprimés en degré.

2. QUELQUES COMMANDES

Fonction	Effet
<code>forward(n)</code>	Avance de n en traçant un trait si le crayon est baissé.
<code>backward(n)</code>	Reculé de n en traçant un trait si le crayon est baissé.
<code>right(n)</code>	Tourne à droite de n degrés.
<code>left(n)</code>	Tourne à gauche de n degrés.
<code>clear()</code>	Efface l'écran.
<code>penup()</code>	Lève le stylo.
<code>pendown()</code>	Baisse le stylo.
<code>reset()</code>	Efface l'écran, remet la tortue au centre et réinitialise ses paramètres
<code>showturtle()</code>	Montre la tortue.
<code>hideturtle()</code>	Cache la tortue.
<code>speed(n)</code>	Change la vitesse de 1(lent) à 10 (rapide). La valeur spéciale 0 est la plus rapide.
<code>update()</code>	Force l'affichage des graphismes en attente.
<code>bye()</code>	Referme la fenêtre.
<code>setup(w,h)</code>	Ouvre une fenêtre de taille w*h .
<code>done()</code>	Evite de fermer la fenêtre.
<code>circle(r)</code>	Trace un cercle de rayon r .
<code>color(cp [,cf])</code>	Couleur du pinceau (cp) [et si existant couleur de remplissage (cf)]
<code>colormode(n)</code>	n =1, ou n =255 dans ce dernier cas on peut utiliser le code (R,V,B)

Pour les couleurs, soit on utilise des constantes prédéfinies (cf. document TurteRef.pdf), soit on peut aussi utiliser un p-tuple de 3 entiers (de 0 à 255) codant la couleur selon le principe du RVB et dans ce cas il faut mettre avant `colormode(255)`.

3. QUELQUES EXEMPLES

Tracé d'un trait :

```
from turtle import *  
  
forward(100)  
  
done()
```



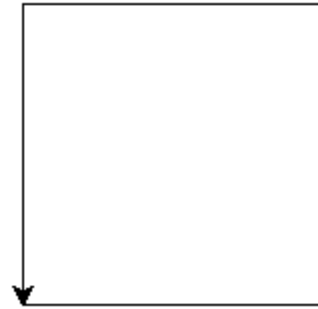
Tracé d'un trait en cachant la tortue :

```
from turtle import *  
  
hideturtle()  
forward(100)  
  
done()
```



Dessin d'un carré :

```
from turtle import *  
  
longueur = 150  
speed(1)  
  
forward(longueur)  
left(90)  
forward(longueur)  
left(90)  
forward(longueur)  
left(90)  
forward(longueur)  
  
done()
```



Dessin d'un carré plein :

```
from turtle import *  
  
longueur = 150  
speed(1)  
  
color('red', 'yellow')  
begin_fill() # début de la zone à remplir  
  
forward(longueur)  
left(90)  
forward(longueur)  
left(90)  
forward(longueur)  
left(90)  
forward(longueur)  
  
end_fill() # fin de la zone à remplir  
done()
```

